Ser321 Principles of Distributed Software Systems

0. Course Objectives, and Required Resources
0.a  Course Objectives and Outcomes

0.a.1  Motivation and Course Outcomes

- **Semester Goals** -- At the end of the semester, you should be able to:
  - Apply the concepts of **concurrency**, **threads** and **synchronization** for use in designing and creating distributed applications; have the ability to create threaded servers and clients using best practices.
  - Utilize **fundamental program development tools** (Linux) -- command-line interpreter, compiler, debugger, build-tools to develop, deploy and test distributed applications consisting of multiple programs and modules.
  - Understand the role **externalization** plays in distributed applications; utilize a language's built-in **serialization** facilities; be able to use binary and text-based externalization, such as **XML** and **Json**.
  - Understand and select among appropriate distribution paradigms for a problem, such as **client-server**, **service-oriented**, and **peer-to-peer**.
  - Utilize common frameworks to design and develop distributed applications, such as **object-based** (**RMI**), **stream-based sockets**, **remote procedure call** (**RPC**).

- Using **C/C++** and **Java** languages; development and deployment on **Linux**
0.b Books, Languages and Development Tools

0.b.1 Ser321 Course Resources

- Course resources can be found online at:

- Textbooks covering most of the course material get either one:
  - Java Network Programming 4th Edition:
  - Java Network Network Programming and Distributed Computing:

- Java Development, For the tutorial see:
  - [http://docs.oracle.com/javase/tutorial/](http://docs.oracle.com/javase/tutorial/)


- Development Environment - Debian Linux. See:

- Deployment Environment - Debian on Raspberry Pi. See: